Customer Management System Cost Analysis Estimate Report

Service Overview

Customer Management System is a fully managed, serverless service that allows you to This project uses multiple AWS services.. This service follows a pay-as-you-go pricing model, making it cost-effective for various workloads.

Pricing Model

This cost analysis estimate is based on the following pricing model: - **ON DEMAND** pricing (pay-as-you-go) unless otherwise specified - Standard service configurations without reserved capacity or savings plans - No caching or optimization techniques applied

Assumptions

- Standard ON DEMAND pricing model for all services
- US East (N. Virginia) region pricing
- Node.js 18.x runtime for Lambda functions
- 128 MB memory allocation for Lambda functions
- Average 500ms execution time per Lambda function
- REST API Gateway (not HTTP API)
- DynamoDB on-demand billing mode
- No caching or optimization applied initially
- Standard customer record size of 1KB

Limitations and Exclusions

- Data transfer costs between regions
- CloudWatch logging and monitoring costs
- Development and testing environment costs
- SSL certificate costs
- Custom domain name costs
- AWS support plan costs
- Third-party integration costs

Cost Breakdown

Unit Pricing Details

Service	Resource Type	Unit	Price	Free Tier
				1M requests/month and 400,000

AWS Lambda	Requests	request	$0.0000002	GB-seconds/month free for first 12 months
AWS Lambda	Compute X86	GB-second (Tier 1)	$0.0000166667	1M requests/month and 400,000 GB-seconds/month free for first 12 months
AWS Lambda	Compute Arm	GB-second (Tier 1, 20% savings)	$0.0000133334	1M requests/month and 400,000 GB-seconds/month free for first 12 months
Amazon API Gateway	Rest Api Tier1	request (first 333M requests)	$0.0000035	1M API calls/month free for first 12 months
Amazon API Gateway	Http Api Tier1	request (first 300M requests, 71% cheaper)	$0.0000010	1M API calls/month free for first 12 months
Amazon DynamoDB	Read Requests	million RRUs	$0.125	25 GB storage always free, 25 RCU/WCU always free
Amazon DynamoDB	Write Requests	million WRUs	$0.625	25 GB storage always free, 25 RCU/WCU always free
Amazon DynamoDB	Storage	GB/month (after first 25GB free)	$0.25	25 GB storage always free, 25 RCU/WCU always free

Cost Calculation

Service	Usage	Calculation	Monthly Cost
AWS Lambda	5 functions handling CRUD operations with 500,000 monthly invocations (Requests: 500,000 requests/month, Compute:	$0.0000002 × 500,000 requests + $0.0000166667 × 32,000 GB-seconds = $0.63	$0.63

	500,000 × 0.5s × 0.128GB = 32,000 GB-seconds/month)		
Amazon API Gateway	REST API handling 100,000 requests per month (Requests: 100,000 requests/month)	$0.0000035 × 100,000 requests = $0.35	$0.35
Amazon DynamoDB	On-demand table with 300,000 reads and 200,000 writes per month, 50GB storage (Reads: 300,000 read request units/month, Writes: 200,000 write request units/month, Storage: 50 GB (25 GB free + 25 GB paid))	$0.125/1M × 0.3M reads + $0.625/1M × 0.2M writes + $0.25 × 25GB storage = $6.41	$6.41
Total	**All services**	**Sum of all calculations**	**$7.39/month**

Free Tier

Free tier information by service: - **AWS Lambda**: 1M requests/month and 400,000 GB-seconds/month free for first 12 months - **Amazon API Gateway**: 1M API calls/month free for first 12 months - **Amazon DynamoDB**: 25 GB storage always free, 25 RCU/WCU always free

Cost Scaling with Usage

The following table illustrates how cost estimates scale with different usage levels:

Service	Low Usage	Medium Usage	High Usage
AWS Lambda	$0/month	$0/month	$1/month
Amazon API Gateway	$0/month	$0/month	$0/month
Amazon DynamoDB	$3/month	$6/month	$12/month

Key Cost Factors

- **AWS Lambda**: 5 functions handling CRUD operations with 500,000 monthly invocations
- **Amazon API Gateway**: REST API handling 100,000 requests per month
- **Amazon DynamoDB**: On-demand table with 300,000 reads and

200,000 writes per month, 50GB storage

Projected Costs Over Time

The following projections show estimated monthly costs over a 12-month period based on different growth patterns:

Base monthly cost calculation:

Service	Monthly Cost
AWS Lambda	$0.63
Amazon API Gateway	$0.35
Amazon DynamoDB	$6.41
Total Monthly Cost	**$7**

Growth Pattern	Month 1	Month 3	Month 6	Month 12
Steady	$7/mo	$7/mo	$7/mo	$7/mo
Moderate	$7/mo	$8/mo	$9/mo	$12/mo
Rapid	$7/mo	$8/mo	$11/mo	$21/mo

- Steady: No monthly growth (1.0x)
- Moderate: 5% monthly growth (1.05x)
- Rapid: 10% monthly growth (1.1x)

Detailed Cost Analysis

Pricing Model

ON DEMAND

Exclusions

- Data transfer costs between regions
- CloudWatch logging and monitoring costs
- Development and testing environment costs
- SSL certificate costs
- Custom domain name costs
- AWS support plan costs
- Third-party integration costs

Recommendations

Immediate Actions

- Switch to ARM-based Lambda functions for 20% compute cost savings
- Implement API response caching to reduce redundant Lambda invocations
- Right-size Lambda memory allocation based on actual performance requirements
- Consider HTTP API instead of REST API for 71% API Gateway cost reduction

Best Practices

- Set up CloudWatch billing alarms for cost monitoring
- Implement DynamoDB single-table design for optimal performance
- Use connection pooling in Lambda functions to reduce cold starts
- Enable DynamoDB auto-scaling for predictable workloads
- Regular monthly cost optimization reviews

Cost Optimization Recommendations

Immediate Actions

- Switch to ARM-based Lambda functions for 20% compute cost savings
- Implement API response caching to reduce redundant Lambda invocations
- Right-size Lambda memory allocation based on actual performance requirements

Best Practices

- Set up CloudWatch billing alarms for cost monitoring
- Implement DynamoDB single-table design for optimal performance
- Use connection pooling in Lambda functions to reduce cold starts

Conclusion

By following the recommendations in this report, you can optimize your Customer Management System costs while maintaining performance and reliability. Regular monitoring and adjustment of your usage patterns will help ensure cost efficiency as your workload evolves.